Serial No. 09/888,273
Response dated January 19, 2006
Office Action dated September 19, 2005

REMARKS/ARGUMENTS

Claims 1, 2, 4-18, 20-26 and 28-30 are pending. Claims 1, 7, 9, 15, 17, 23, and 25 have been amended to put these claims into better form. Claims 1, 2, 7, 9-11, 15, 17, 18, 23, 25 and 26 are rejected under 35 U.S.C. 102(e) as being anticipated by Sager, US Patent 6,542,921 ("Sager"). Applicants gratefully acknowledge the Office Action's indication that claims 4-6, 8, 12-14, 16, 20-22, 24 and 28-30 would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Applicants respectfully submit nowhere in Sager is the disclosure, teaching or suggestion of "[a] method of assigning thread priority comprising: ... determining if there is an indication of approaching instruction side starvation for said first thread; and incrementing a value stored in said first starting counter in response to an indication of approaching instruction side starvation for said first thread" (e.g., the embodiment as recited in amended claim 1).

First, the Office Action asserts Sager has taught determining if there is an indication of approaching instruction side starvation for said first thread in the Abstract and the description of Figure 11, element 1117. Applicants disagree. The Abstract states:

> The present invention provides a method and apparatus for controlling a processing priority assigned alternately to a first thread and a second thread in a multithreaded processor to prevent deadlock and livelock problems between the first thread and the second thread. In one embodiment, the processing priority is initially assigned to the first thread for a first duration. It is then determined whether the first duration has expired in a given processing cycle. If the first duration has expired, the processing priority is assigned to the second thread for a second duration.

11

Serial No. 09/888,273
Response dated January 19, 2006
Office Action dated September 19, 2005

The Abstract merely describes a general intention to address deadlock and livelock

problems between a first and second thread. It further describes that when a first duration

of processing priority is expired, a second duration to a second thread is assigned. There

is no mention of at least "...*determining if there is an indication of approaching*

instruction side starvation for said first thread ..." as specifically recited in the

embodiment of claim 1.

> Next, the description of Element 1117 states:

> At decision block 1117, the process proceeds to block 1121 if it is
> determined that the processing priority has been switched from thread 0 to
> thread 1 in the current processing cycle and loops back to block 1105
> otherwise. In one embodiment, whether the processing priority has been
> switched back from thread 0 to thread 1 in the current processing cycle
> can be determined by detecting a signal indicating that the content of the
> TPC has reached the predetermined threshold value in the current
> processing cycle and that the TPB has been inverted from 0 to 1 in the
> current processing cycle. The determination of whether the processing
> priority has been switched from thread 0 to thread 1 in the current
> processing cycle will be described in more detail below.

The above cited section describes the switching of a thread 0 to a thread 1 conditioned

upon the detecting a signal indicating a predetermined threshold value and another the

inversion of another value. However, again there is no mention of at least the specifically

claimed limitation "...*determining if there is an indication of approaching instruction*

side starvation for said first thread ..." as found in the embodiment of claim 1.

Applicants submit that these cited sections do not address Iside starvation or

incrementing values at all, and that the Office Action's assumption regarding the

operation of Sager are unsupported by the text of Sager reference.

> Furthermore, Applicants submit the cited reference does not teach

"...*incrementing a value stored in said first starting counter in response to an indication*

12

Serial No. 09/888,273
Response dated January 19, 2006
Office Action dated September 19, 2005

of approaching instruction side starvation for said first thread... " as recited in the

embodiment of claim 1. The Office Action cites Figure 11, element 1121 and column 9,

lines 12-29 in its rejection.

The description of element 1121 states:

> At block 1121, the TC0 content is incremented by a predetermined number, for
> example 1. The content of the TC0, as explained above, will be used to load into
> the TPC to indicate how long the priority duration for thread 0 will be the next
> time thread 0 is given the processing priority.

Applicants submit that the cited section describes the incrementing of TC0 content.

However, column 12 lines 1-3 state:

> T0 counter (TC0) is used to hold a value that corresponds to a duration for which
> the thread 0 is given the processing priority.

Its is clear that the incrementing of TC0 counter found in the cited section above has

nothing to do with approaching iside starvation, but rather to determining *the duration of*

time for which a thread is to be give processing priority. Next, column 9, lines 12-29

state:

> As each thread is being executed, its progress is monitored to determine whether
> it is being stuck. *If a particular thread, for example thread 0, has not made any*
> *progress in the period of time during which it has priority then it will be given*
> *priority for a longer duration of time the next time it has priority.* This duration of
> time during which thread 0 is given priority will continue to increase until thread
> 0 makes progress. Once it is determined that thread 0 has made progress, its
> priority duration can be reset to some shorter period, for example the initial
> duration. Likewise, the duration of time during which thread 1 is given priority
> will continue to increase until thread 1 makes some progress at which time its
> priority duration can be reset to some shorter period, for example the initial
> duration. *In short, the length of time during which each thread has priority will*
> *continue to increase until that particular thread makes some progress. (emphasis*
> *added)*

13

Applicants submit this section reflects the operation of the TC0 counter described above, in that it addresses the appropriate amount of time a thread is to be given priority, to be determined based upon the amount of progress the given thread has made. Again, this section does not address "...*incrementing a value stored in said first starting counter in response to an indication of approaching instruction side starvation for said first thread...*" as described in the embodiment of claim 1.

Therefore, Applicants respectfully submit since the cited reference does not disclose multiple limitations as recited in the embodiment of claim 1, it is inadequate to support a proper 35 U.S.C. § 102(e) rejection. Independent claims 17 and 25 contain similar allowable limitations. Dependent claims 2-6, 18-22 and 26-30 are allowable for depending from allowable base claims.

Furthermore, Applicants' respectfully submit that nowhere is the disclosure, teaching or suggestion of "[a] method of assigning thread priority comprising ... assigning priority to a second thread in response to one of a plurality of conditions being true, the conditions consisting of... if there is not an indication of approaching instruction side starvation for said first thread." (e.g., the embodiment as recited in amended claim 7).

The Office Action asserts that Sager has taught the claimed limitations at Figure 9 (elements 913 and 917). The Office Action further states that when the priority of the second thread is assigned in response to only the current priority period expiring, this means that the condition is true that there is not an indication of approaching instruction side starvation for said thread. Applicants disagree.

14

Serial No. 09/888,273
Response dated January 19, 2006
Office Action dated September 19, 2005

First, Applicants first reiterate all arguments made above regarding Iside

starvation. Moreover, Applicants' respectfully submit that Office Action's assumptions

regarding Sager's disclosure of assigning priority based on a plurality of conditions

including indication of approaching Iside starvation are both unsupported and not taught

suggested or disclosed in the Sager reference. The description of elements 913 and 917

of Sager merely state:

> At decision loop 913, the method 900 proceeds to block 917 if the current
> priority duration has expired. At block 917, the processing priority is alternated,
> i.e., assigned to the other thread.

The cited section merely describes the operation

It is clear Sager does not disclose assigning priority based on a plurality of

conditions including no indication of approaching Iside starvation as described in the

embodiment of claim 7 in its description of elements 913 and 917. The Office Action's

assumptions are insufficient and must be found in the reference to form the basis of a

proper 35 U.S.C. 102(e) rejection. In order to be a proper 35 U.S.C. 102(e) rejection, the

claimed limitations must be taught, disclosed or suggested in the cited reference, which

they are not. Independent claims 15 and 23 contain substantively similar claim elements,

and therefore are allowable as well. Dependent claims 8, 16 and 24 depend from

allowable base claims and therefore should be allowed as well.

For at least all the above reasons, the Applicants respectfully submit that this

application is in condition for allowance. A Notice of Allowance is earnestly solicited.
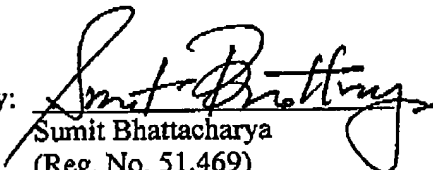
15

Serial No. 09/888,273
Response dated January 19, 2006
Office Action dated September 19, 2005

The Examiner is invited to contact the undersigned at (408) 975-7500 to discuss

any matter concerning this application. The Office is hereby authorized to charge any

additional fees or credit any overpayments under 37 C.F.R. § 1.16 or § 1.17 to Deposit

Account No. **11-0600**.

Respectfully submitted,
KENYON & KENYON

Dated: <u>January 19, 2006</u>   By: _____
Sumit Bhattacharya
(Reg. No. 51,469)
Attorneys for Intel Corporation

KENYON & KENYON
333 W. San Carlos St., Suite 600
San Jose, CA 95110
Telephone: (408) 975-7500
Facsimile: (408) 975-7501

81117.1

16